Message-ID: <bafccde7f62b1a19b481095f8f57eccb@alexburke.ca>
Date: Tue, 01 Nov 2022 21:28:34 +0000
From: alex@...xburke.ca
To: oss-security@...ts.openwall.com
Subject: Re: OpenSSL X.509 Email Address 4-byte Buffer
 Overflow (CVE-2022-3602), X.509 Email Address Variable Length Buffer
 Overflow (CVE-2022-3786)

>When will OpenSSL be replaced by something written in a safe language [...]?

I fully expect this to never occur, but the optimist in me hopes that someone will prove me wrong.

C is a hell of a drug.

Cheers,
Alex

November 1, 2022 7:13 PM, "Demi Marie Obenour" <demi@...isiblethingslab.com> wrote:

> On Tue, Nov 01, 2022 at 06:08:34PM +0100, Solar Designer wrote:
>
>> OpenSSL Security Advisory [01 November 2022]
>> ============================================
>>
>> X.509 Email Address 4-byte Buffer Overflow (CVE-2022-3602)
>> ==========================================================
>>
>> Severity: High
>>
>> A buffer overrun can be triggered in X.509 certificate verification,
>> specifically in name constraint checking. Note that this occurs
>> after certificate chain signature verification and requires either a
>> CA to have signed the malicious certificate or for the application to
>> continue certificate verification despite failure to construct a path
>> to a trusted issuer. An attacker can craft a malicious email address
>> to overflow four attacker-controlled bytes on the stack. This buffer
>> overflow could result in a crash (causing a denial of service) or
>> potentially remote code execution.
>>
>> Many platforms implement stack overflow protections which would mitigate
>> against the risk of remote code execution. The risk may be further
>> mitigated based on stack layout for any given platform/compiler.
>>
>> Pre-announcements of CVE-2022-3602 described this issue as CRITICAL.
>> Further analysis based on some of the mitigating factors described above
>> have led this to be downgraded to HIGH. Users are still encouraged to
>> upgrade to a new version as soon as possible.
>>
>> In a TLS client, this can be triggered by connecting to a malicious
>> server. In a TLS server, this can be triggered if the server requests
>> client authentication and a malicious client connects.
>>
>> OpenSSL versions 3.0.0 to 3.0.6 are vulnerable to this issue.
>>
>> OpenSSL 3.0 users should upgrade to OpenSSL 3.0.7.
>>
>> OpenSSL 1.1.1 and 1.0.2 are not affected by this issue.
>>
>> This issue was reported to OpenSSL on 17th October 2022 by Polar Bear.
>> The fixes were developed by Dr Paul Dale.
>>
>> We are not aware of any working exploit that could lead to code execution,
>> and we have no evidence of this issue being exploited as of the time of

```
>> release of this advisory (November 1st 2022).
>>
>> X.509 Email Address Variable Length Buffer Overflow (CVE-2022-3786)
>> ==================================================================
>>
>> Severity: High
>>
>> A buffer overrun can be triggered in X.509 certificate verification,
>> specifically in name constraint checking. Note that this occurs after
>> certificate chain signature verification and requires either a CA to
>> have signed a malicious certificate or for an application to continue
>> certificate verification despite failure to construct a path to a trusted
>> issuer. An attacker can craft a malicious email address in a certificate
>> to overflow an arbitrary number of bytes containing the `.' character
>> (decimal 46) on the stack. This buffer overflow could result in a crash
>> (causing a denial of service).
>>
>> In a TLS client, this can be triggered by connecting to a malicious
>> server. In a TLS server, this can be triggered if the server requests
>> client authentication and a malicious client connects.
>>
>> OpenSSL versions 3.0.0 to 3.0.6 are vulnerable to this issue.
>>
>> OpenSSL 3.0 users should upgrade to OpenSSL 3.0.7.
>>
>> OpenSSL 1.1.1 and 1.0.2 are not affected by this issue.
>>
>> This issue was discovered on 18th October 2022 by Viktor Dukhovni while
>> researching CVE-2022-3602. The fixes were developed by Dr Paul Dale.
>>
>> We have no evidence of this issue being exploited as of the time of
>> release of this advisory (November 1st 2022).
>>
>> References
>> ==========
>>
>> URL for this Security Advisory:
>> https://www.openssl.org/news/secadv/20221101.txt
>>
>> Note: the online version of the advisory may be updated with additional details
>> over time.
>>
>> For details of OpenSSL severity classifications please see:
>> https://www.openssl.org/policies/secpolicy.html
>
> 1. Why OpenSSL is even *parsing* these SANs? In TLS they will never be
> used, so parsing them is not just extra attack surface but also a
> waste of resources. I understand that parsing them is important for
> S/MIME, but that does not mean OpenSSL should *always* parse them.
> Instead, OpenSSL should only parse them when a certificate needs to
> be verified against an email address, which TLS never requires.
>
> 2. Why was this not caught by fuzzing? Is this code not fuzzed for some
> reason?
>
> 3. When will OpenSSL be replaced by something written in a safe
> language, or at least with a better-maintained fork? I know that
> distributions often cannot use LibreSSL (because FIPS, ugh) or
> BoringSSL (because of no stable API or ABI), but I wonder if e.g.
> libcurl should be linked to BoringSSL instead.
> --
> Sincerely,
> Demi Marie Obenour (she/her/hers)
> Invisible Things Lab
```